

# Visual Exploration of Multivariate Graphs

Martin Wattenberg

Visual Communication Lab, IBM Research

1 Rogers St., Cambridge MA 02142

[mwatten@us.ibm.com](mailto:mwatten@us.ibm.com)

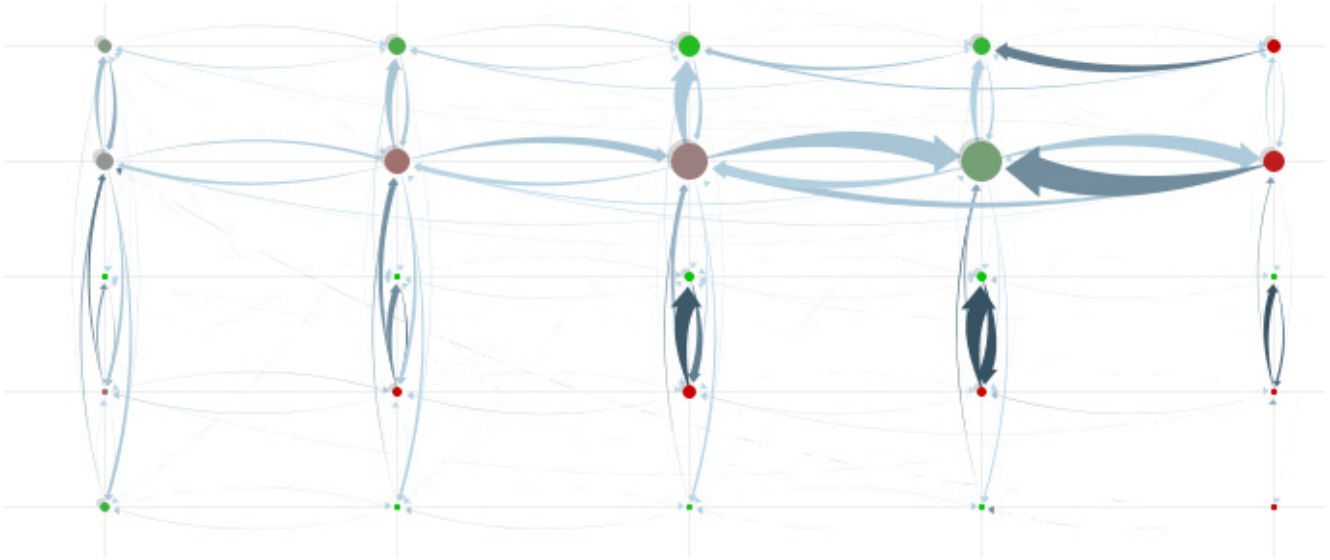


Figure 1. A PivotGraph visualization of a large graph rolled up onto two categorical dimensions

## ABSTRACT

This paper introduces PivotGraph, a software tool that uses a new technique for visualizing and analyzing graph structures. The technique is designed specifically for graphs that are “multivariate,” i.e., where each node is associated with several attributes. Unlike visualizations which emphasize global graph topology, PivotGraph uses a simple grid-based approach to focus on the relationship between node attributes and connections. The interaction technique is derived from an analogy with methods seen in spreadsheet pivot tables and in online analytical processing (OLAP). Finally, several examples are presented in which PivotGraph was applied to real-world data sets.

## Author Keywords

information visualization, graph drawing

ACM Classification Keywords

H.5.2. [Information Interfaces]: User Interfaces. I.3.6 [Methodology and Techniques]: Interaction Techniques

## INTRODUCTION

A common goal in exploratory data analysis is to form hypotheses about a graph. In contexts ranging from social networks to Markov chains, it can be important to understand the properties of a graph and make inferences about why it looks the way it does. Because of the complex structure and large size of many graphs, such an analysis can be difficult.

Visualization is one solution to this problem. In one popular approach, nodes of a graph are represented by a dot, and edges by lines or curves linking them. Such “node-and-link diagrams” have been explored intensively throughout this century (see [3, 10, 14, 15]). A second common approach is a matrix view, or density table, that displays a graph’s adjacency matrix overlaid on a grid [4].

Both node-and-link diagrams and matrix views emphasize link structure. In many cases, however, data about a graph

includes additional information, such as properties of individual nodes. For example, a graph where nodes represent people may be accompanied by details about each person’s race, gender, and age. Such attributes may be continuous (such as age) or discrete and categorical (e.g., race or gender). Such a structure will be called a *multivariate graph* in this paper. Each property that a node might have (again, take race or gender in a social network) will be called a *dimension* of the graph. This paper focuses specifically on visualizing multivariate graphs where the nodes have several discrete categorical dimensions.

Because multivariate graphs are so common in practice, many popular graph visualization tools (e.g., [1, 2, 6, 17, 18]) include ways of displaying them. The most common technique is to use a standard node-and-link layout but to draw nodes differently based on their values; often attribute values are associated with different colors or shapes. A second method is to use a matrix view, and to group the positions of nodes on the axes based on their values in one or more dimensions.

Although both techniques are natural extensions of familiar displays, each has drawbacks. Colored node-and-link diagrams are poor for quantitative comparisons between groups. Answering a question such as “who has more connections, men or women?” requires collating information that is potentially spread all over the diagram. For an analyst who wishes to see the interaction between two variables—e.g., how race affects patterns of communication between genders—matrix views quickly become unwieldy since the axes must be sorted on two variables at once.

This paper presents a software tool called PivotGraph that makes multidimensional comparisons straightforward. The technique introduced in PivotGraph has the potential to be an effective complement to traditional graph topology diagrams. It may be of interest to the HCI community both as a new visualization and interaction technique, and as a practical tool for those who wish to analyze graphs.

Section 2 describes the basic method used in PivotGraph along with motivation and related work. Section 3 describes the software itself along with a detailed discussion of the design rationale. Section 4 describes a series of three pilots in which the software was tested by subject-matter experts on real-world data sets they had previously analyzed, to see whether they found patterns they had previously missed. Section 5 discusses a set of natural extensions and future research directions.

### ANALYZING MULTIVARIATE GRAPHS

PivotGraph is designed to highlight interactions between the various dimensions of a graph, and was motivated in part by previous work on displaying other kinds of multivariate data.

One popular framework for exploring multivariate data is Online Analytical Processing (OLAP) [7]. The underlying

structure used in OLAP is a “data cube,” a set of items which have values on various dimensions. The classic example of a data cube is a collection of sales transactions where each sale is associated with a product, a store, a salesperson, and so on.

OLAP has two key reporting operations. The first is *roll-up*, a summarization query of the form “Show me total sales of each product” or “Show me total sales for each product/store combination.” The second is *selection*, where a user can ask for information on a subset of items; typically selections are combined with roll-ups in queries such as “Show me the total sales of product X at each store.” Users who are more familiar with spreadsheets than databases may recognize these as the same techniques used in a pivot table.

One of the benefits of roll-up and selection operations is that they reduce the number of dimensions of a data set. A natural question is whether similar techniques could be used to transform a large multivariate graph into a simpler summary graph that is easy to visualize and manipulate. In fact, it is straightforward to apply the OLAP operations to multivariate graphs; the nodes already form a data cube, and there is a natural extension for edges. This generalization is described below.

### Roll-up and Selection for Multivariate Graphs

Consider graphs which have weighted, possibly directed edges, and whose nodes have values in a set of categorical dimensions, as well as a notion of “size”. (The concept of node size becomes helpful when aggregating nodes.)

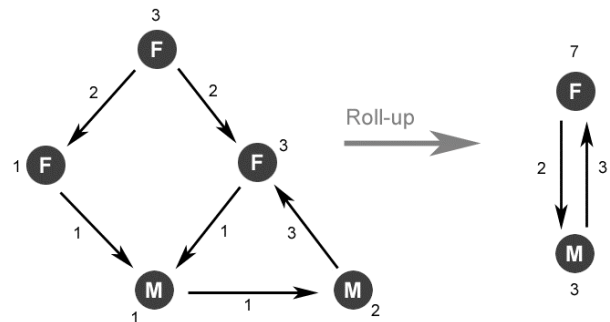


Figure 2. Roll-up of a simple social network on “gender” dimension. Numbers indicate weights of edges and sizes of nodes.

The graph-theoretic analogue of the roll-up operation is a combination of classical node aggregation and edge contraction techniques. Informally, to roll up a graph on a set of dimensions, one simply aggregates all nodes that have the same values on each of those dimensions, and aggregate edges accordingly.

More formally: Given an initial graph  $G$  and a set of dimensions  $D$ , the roll-up graph  $R_D(G)$  is defined as

follows. Divide the nodes of  $G$  into a set of  $n$  equivalence classes  $S = \{S_1, S_2, \dots, S_n\}$  where two nodes are equivalent if they have same value on each dimension in  $D$ . For  $1 \leq i \leq n$ , define a node  $N_i$  of  $R_D(G)$ , with the size of  $N_i$  set to the sum of the sizes of nodes in  $S_i$ . Finally, for  $1 \leq i \leq n$ ,  $R_D(G)$  is defined to have an edge between  $N_i$  and  $N_j$  if  $i \neq j$ , and there are any edges in  $G$  between the nodes in  $S_i$  and  $S_j$ . (In some contexts this definition might be modified to allow self-loops, that is allow  $i = j$  in the above definition.) The weight of the edge between  $N_i$  and  $N_j$  will be the sum of the weights of all edges between nodes in  $S_i$  and  $S_j$ .

An analogue of selection is a simpler operation that results an *induced subgraph* [5]. The idea is to take a subset of the original graph whose nodes have specified values on a particular set of dimensions. Figure 3 gives an example; formally, for a set of  $m$  dimensions  $D = \{D_1, \dots, D_m\}$  and  $m$  values  $\{V_1, \dots, V_m\}$ , the nodes of selection graph  $S_{D,V}(G)$  are all the nodes  $N$  of  $G$  such that  $D_i(N) = V_i$  for each  $1 \leq i \leq m$ . The edges of  $S_{D,V}(G)$  are all edges of  $G$  that connect nodes of  $S_{D,V}(G)$ .

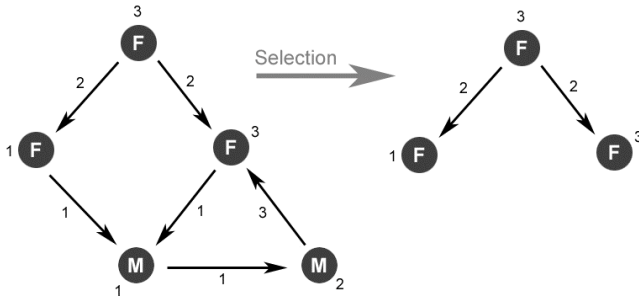


Figure 3. Selection on “gender=female”

### Visualizing Graphs with Few Data Dimensions

The roll-up and selection transformations both simplify a graph by removing dimensions from consideration, and in the process they typically reduce the number of nodes and edges as well. As discussed above, one of the motivations for using them here is that simplifying a graph to fewer dimensions makes it easier to visualize. In particular, once a graph has been reduced to two dimensions,  $D_1$  and  $D_2$ , it has a natural scatterplot-like layout. Each value of  $D_1$  is assigned a position on the  $x$ -axis, and each value of  $D_2$  is assigned a position on the  $y$ -axis. Then a given node in  $G$  is assigned  $x$  and  $y$  coordinates based on its values on dimensions  $D_1$  and  $D_2$ . Figure 4 shows an example of this method. At left is a standard node and link diagram showing an undirected social network where nodes are classified by gender (M or F) and company division (1 or 2). At right is a scatterplot-style “PivotGraph diagram”, based on a roll-up on the gender and division dimensions. Note that in the roll-up view, node size corresponds to the number of nodes being aggregated, and edge thickness corresponds to the number of edges being aggregated.

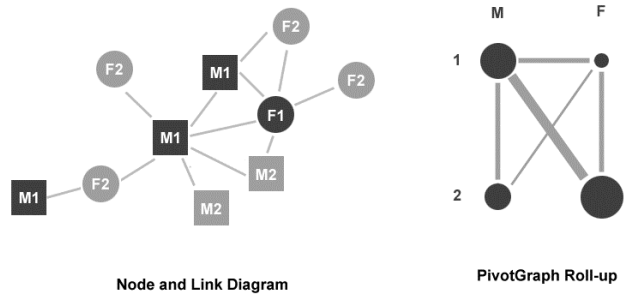


Figure 4. Node-and-link diagram versus PivotGraph

The two types of diagrams have different strengths. While the node-and-link diagram makes the topology clear, the PivotGraph diagram makes it immediately obvious that there are connections between all gender/division pairs, with the exception of men and women in division 2. The PivotGraph diagram uses circle size and line thickness to show node sizes and edge weights, allowing quantitative comparisons and conveying a sense of salience of nodes and edges.

### Related work

Although the combination of a grid-based layout with these OLAP-style operations is new, the PivotGraph technique is related to several streams of previous work.

Aggregation of nodes and edges according to various criteria is a well-known technique for simplifying complex graphs [3, 15], and there has been recent work on the idea of letting the user dynamically control the level of detail by aggregating based on topological criteria [13]. Systems such as Pajek [2], Tulip [1] and the InfoVis Toolkit [8] also have some degree of support for such computations. Many social network analysis tools allow some form of selection [6] and some packages, such as NetMiner [18] and InFlow [17] can compute roll-ups on a single dimension.

The idea of placing nodes according to their properties (rather than edge topology) appeared in the work of Fekete and Wang [9]. Instead of working with a grid, however, they used a treemap to lay out nodes taking values in a hierarchy. NetDraw [6] has a layout option based on node attributes but does not perform a roll-up operation.

PivotGraph is not the first visualization technique to depend on the OLAP paradigm. Gray *et al.* [12] describe a set of data-cube operators with the purpose of supporting visualization, though they do not discuss particular visualization techniques. The Polaris system [20], which led to the commercial Tableau product [22], has a sophisticated framework for visualizing traditional data cubes, although they do not treat graph-based data. Finally, when rolled-up on one dimension, the PivotGraph layout is reminiscent of both the ThreadArcs visualization used for email threads in [16] and Arc Diagrams [24].

## THE PIVOTGRAPH APPLICATION

This section describes the PivotGraph software that allows users to interactively explore roll-ups and selections of multivariate graphs. PivotGraph is a desktop application written in Java. See Figure 5 for a screenshot. (Note that in this and several other screenshots some text was changed in order to mask confidential data.) The PivotGraph interface has three components. A traditional menu bar lets users handle files and change various viewing parameters. At the left is a panel with three parts: two drop-down menus to determine roll-up dimensions for the  $x$ - and  $y$ -axes; a legend; and a set of drop-down menus, one for each dimension, that allow the user to specify selection parameters. Finally, the bulk of the screen is devoted to the graph visualization itself.

### Visualization

Although the basic idea of the scatterplot representation described in the previous section is simple, it turns out there are a number of subtle challenges that need to be addressed for the visualization to remain legible.

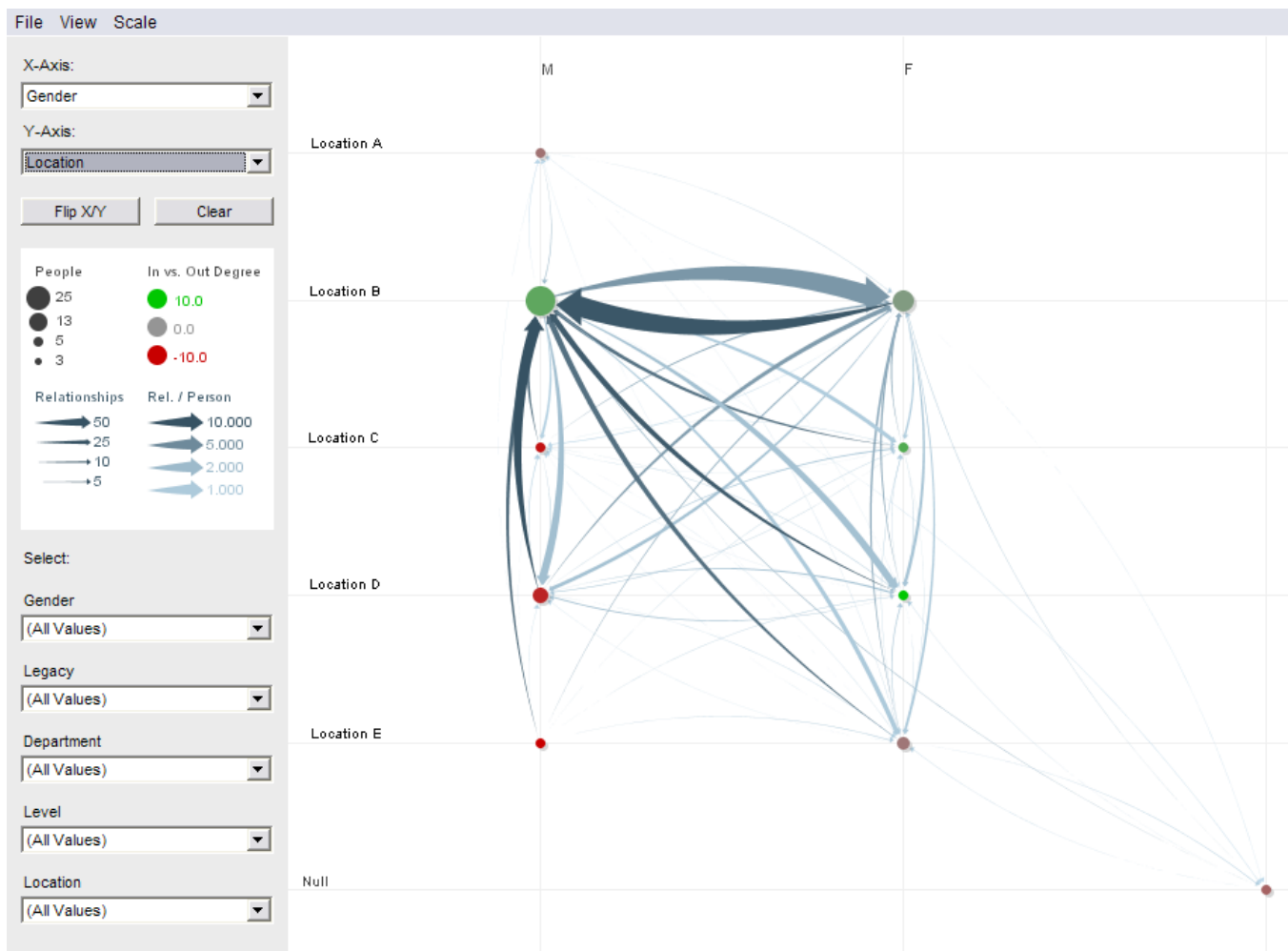
Before diving into the details, it may be helpful to describe the data shown in Figure 5, which is a good example of the

Figure 5. *Screenshot of PivotGraph in action.*

visualization at work. The screenshot shows an anonymized view of a real social network within a corporation. Nodes in the graph represent people, and edges represent communication. The graph is rolled up by gender ( $x$ -axis) and office location ( $y$ -axis). Several patterns can be seen in the visualization. There is a large amount of cross-gender communication in Location B, for example, but very little elsewhere. Men in Location B seem to be especially central, with women in locations C, D, and E communicating more with them than with men in their own locations. The node sizes provide an indication of how many men and women are at each location, and it is easy to see that in the graph one location (A) has only men.

### Layout

Each node is represented by a circle whose  $x$ - and  $y$ -coordinates are determined by the current roll-up dimensions. (If there is only one roll-up dimension, then the dots are laid out on a line, as in figure 6.) The area of each circle is proportional to the size variable of the node. In an early version of the program, the sequence of dimension values on the axes was determined simply by alphabetical ordering. A second version of the program rearranged the order of the values to create a more meaningful use of space. To do this for a given dimension, the roll-up of the graph onto that dimension is created, after which a





barycentric heuristic [3, p.103] is applied to arrange the nodes. This can help reduce clutter and create a more natural order, since highly connected nodes tend to be placed near each other.

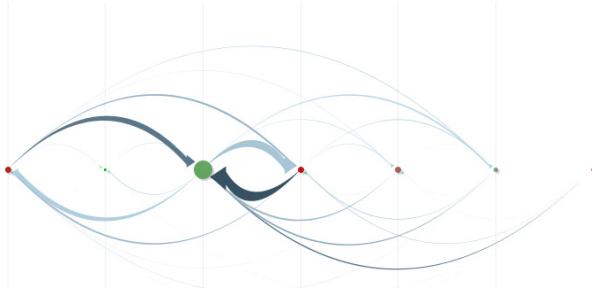


Figure 6. *A one-dimensional layout*

Edges are drawn between nodes, with widths corresponding to their weights. When the edges are directed, they are given arrowheads. Because of the grid structure of the graph layout, straight edges would almost always have serious occlusion problems. To avoid this, the program draws the edges as quadratic curves with a slight bend. In a one-dimensional roll-up, where all nodes are in a line, the bend angle is increased. When the edges are drawn, they are rendered in order from thinnest to thickest, so that the largest (usually most important) edges are less likely to be occluded.

### Colors

Both nodes and edges are color-coded. The nodes can be colored according to any quantitative attribute; this could correspond either to measured data (such as age of a person in a social network) or derived statistics (such as the difference between node in-degree and out-degree).

Edges are also colored. In theory the color could be tied to any measurement, just as for nodes, but in practice it seems that one particular is especially useful. In this color scheme, darkness indicates the ratio of the weight of a directed edge to the size of the originating node. With many data sets this statistic turns out to correspond to an intuitive notion of salience. For example, in a graph that is formed by rolling up an initial graph of equally weighted nodes and equally weighted edges, this statistic describes the average number of edges per originating node. In comparisons that involve two dimensions, and where the rolled-up nodes have different sizes, this normalized statistic can be critical.

### Interaction

PivotGraph allows several types of interactions, which are an integral part of the system. The user can choose roll-up dimensions for the  $x$ - and  $y$ -axes from drop-down menus, and use drop-downs to make a graph selection. Currently roll-ups are limited to two dimensions, but selections can be defined on any number of attributes. As roll-up and selection parameters are chosen, the visualization changes immediately in response. The user may also move the

mouse over both nodes and edges for additional detailed information. As with direct-manipulation analysis tools such as Spotfire [19], the ability to rapidly modify views and drill down for details enhances the power of the underlying visualizations.

A critical aspect of the interactivity of PivotGraph is animation of the transitions when the user changes roll-up axes. Instead of flipping directly from one roll-up to another, an animation smoothly interpolates between the two roll-ups. Because usage of PivotGraph often involves moving between several different views, keeping a sense of orientation is a high priority. The transitions were designed to clarify the action of switching roll-up dimensions by providing users with the sense that they are smoothly navigating through a unified information space. Both informally and in the pilot tests described below, users were extremely enthusiastic about this feature. (Indeed, the name PivotGraph stems partly from an analogy with spreadsheet pivot tables but mainly from the strong feeling of spatial rotation that turned out to be central to the user experience.)

The technical implementation of the transitions is not completely straightforward. One issue is that to interpolate between two different 2-dimensional roll-ups can require an intermediate representation showing a roll-up of 3 or 4 dimensions (the union of the two sets of axis dimensions). For instance, consider a graph  $G$  with three dimensions  $A$ ,  $B$ , and  $C$ . If  $A$  and  $B$  are the current axis dimensions, and the user switches to showing  $A$  and  $C$ , perform the following steps:

1. Calculate a transitional graph  $T$  by rolling up  $G$  on  $(A, B, C)$ .
2. Display  $T$  by choosing coordinates for nodes based on the dimensions  $A$  and  $B$ . (This step will typically involve overplotting of nodes and edges, leading to a graph that looks similar but not identical to the roll-up on  $(A, B)$ ).
3. Use linear interpolation to animate to a view of  $T$  where coordinates of nodes are based on dimensions  $A$  and  $C$ .
4. Switch to a view of the graph created by rolling up onto  $A$  and  $C$  alone.

Because there can potentially be hundreds of edges, each one of which is a complex path of lines and quadratic curves, maintaining a reasonable frame rate (15-20 frames per second on a 1 GHz Windows PC) and visual simplicity requires some compromises. First, during the animation the edges are drawn without anti-aliasing. Second, and more important, is that the thinnest edges are not rendered during the transition. It turns out that drawing just a few of the most prominent edges is enough to maintain a sense of orientation and continuity.

The tool was designed tool for rapid exploration, with the goal of encouraging easy movement between different

views. Both in informal observations and in the interviews described below, users often took advantage of this interactivity.

### Comparison with a Matrix View

Figure 4 shows a very simple comparison of PivotGraph with a node-and-link diagram. One might also ask how PivotGraph compares to an analogous matrix view. As described in [4], a matrix view of a graph with  $n$  nodes displays an  $n \times n$  grid, with the cell at  $(i, j)$  colored to show the strength of the connection between node  $i$  and node  $j$ . Matrix views allow a comprehensive view of all individual connections and it has been recently suggested that for some tasks they outperform node-and-link views [11].

To form a matrix view of a graph rolled up on two dimensions, one typically sorts the nodes by value on the first dimension and then, for nodes with the same value, sorts by the value on the second dimension as a tie-breaker. Figure 7 shows such a matrix view of a synthetic graph rolled up on dimensions of “color” and “taste.” (Strength of connection is shown by darkness of cell shading.) Figure 8 shows the PivotGraph view of the same data. While the matrix view could be helpful to an experienced user, its emphasis on the first sort dimension (“color”) hides some information. For example, in both figures it is easy to see that there are no connections between “green” nodes. On the other hand, in the PivotGraph view it is equally easy to see no “sour” nodes are connected, while the matrix view makes this fact somewhat obscure. Thus for inspecting certain relations on two dimensions at once, the PivotGraph view may be preferable.

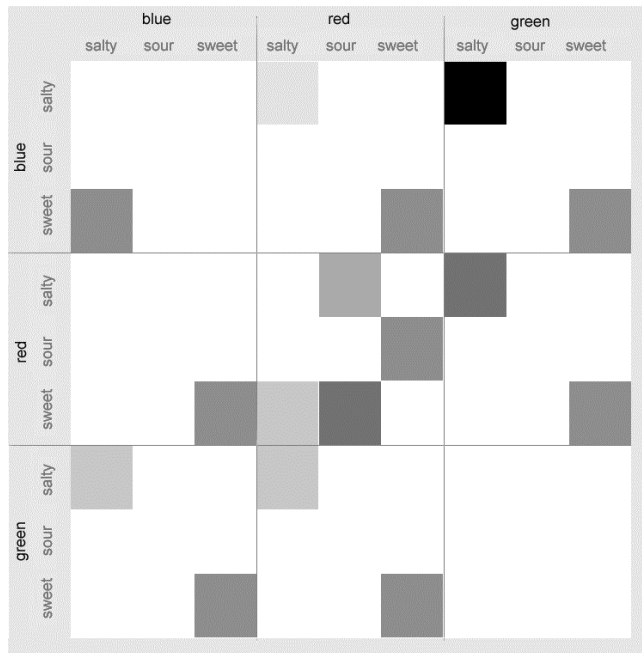


Figure 7. Matrix diagram of a synthetic graph.

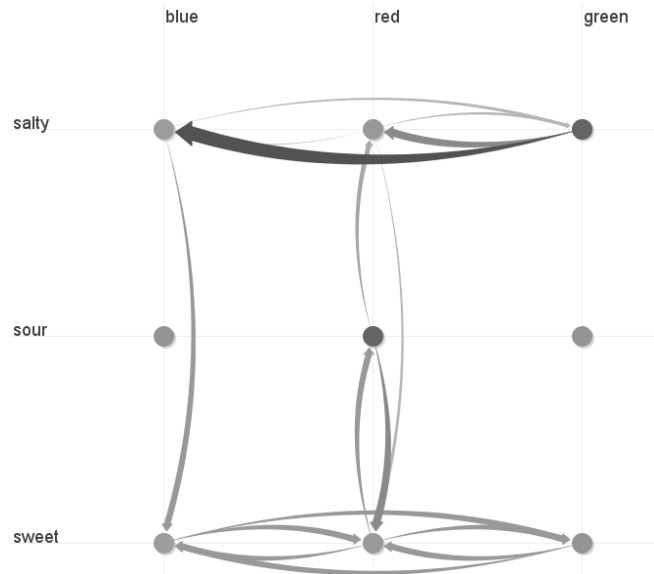


Figure 8. PivotGraph view of the graph in Figure 7.

### Scope

Although this paper argues that PivotGraph represents a useful tool, the technique has some limitations in scope that should be noted. In particular, several aspects of graph topology are not preserved under roll-up and selection transformations. For example, after roll-up a disconnected graph may become connected and an acyclic graph may become cyclic. Selection can also change global properties such as connectedness. Thus a traditional node-and-link diagram is preferable when analyzing topological properties. PivotGraph diagrams cannot replace traditional methods, but rather should be used alongside them as a complementary technology.

A second limitation comes from the method of ordering category values on the axes. The technique described above does not always produce perfect results, and can potentially be slow for very large graphs. As described in the pilot usage below, it may be desirable to allow users to change the ordering by hand.

A final restriction is that PivotGraph can use at most two dimensions for coordinates. This does not mean, however, that it is limited to analysis of effects that involve one or two variables. More complex effects can be explored through use of the selection operator and by interactively changing the choice of roll-up axes. In this way PivotGraph is analogous to a tool like Spotfire, in which the user manipulates a 2D scatterplot view on high-dimensional data. The section on future directions discusses potential techniques for displaying additional dimensions directly.

### PILOT USAGE

PivotGraph has been applied to a range of data sets. This section describes feedback from three pilot uses of PivotGraph, in which it was given to a total of five analysts who were encouraged to look for new patterns in their own

data. In each case, the users had employed other sophisticated tools for previous analyses. Since the goal of PivotGraph is to help users find new patterns and spot undiscovered features of graph data, one test of its value was to watch experts analyze familiar data to see if they would find previously unknown facts and patterns.

For each of the three pilots, each analyst was given a copy of the program which automatically loaded their data. The author then engaged in a semi-structured interview in which the program was first described, and then the analyst was asked to spend some time using it while they explained what they were looking for and what patterns they were finding. In each pilot a subject found patterns which they had not been aware of before. All users reported a positive response to the tool; while this is potentially due to a novelty effect or desire to please the interviewer, in several cases users took actions that backed up their reports. Specifics are below, although names have been changed and details have been removed for reasons of privacy and confidentiality.

### Pilot 1

The initial motivation for the creation of PivotGraph came from a project that involved analysis of a transition matrix consisting of 521 states (nodes) and 2,671 transition probabilities (weighted edges). Each state had four associated categorical attributes.

The group studying this data had worked with it for several months and used custom-built database reporting and charting tools for viewing and analysis. In conversations with the author they expressed interest in having a new way of visualizing the data. The first version of PivotGraph was created in response to this interest. Three of the people on the project ran PivotGraph: Allen, a computer scientist; Deanna, a software developer, and Bob, a senior executive. Each was interviewed separately.

Bob, the executive, began trying different x- and y-axis combinations as soon as he was shown the tool. He pointed to several cross-variable trends, including one he termed “weird and interesting.” Bob said he had not previously spotted any of these patterns, despite having “stared” at the data before, and said he definitely wanted to use the program again. During the session Bob continued to change the view configuration and made several positive comments on the animated transitions.

Deanna, the developer, said that she saw “totally different” patterns in the data than she had seen before. As the keeper of the database that contained the transition information, she had spent a significant amount of time handling the data. Like Bob, she made many cross-variable comparisons. She too said she would like to use the program again.

When Allen, the computer scientist, began using PivotGraph, he quickly pointed to a series of anomalous transitions. In one roll-up (Figure 9) a certain feature was so implausible that it seemed likely to be caused by a bug in

the PivotGraph code. After careful checking, however, it was confirmed that the anomaly really was present in the data—a finding which led to subsequent action on the part of the group. Thus the software had uncovered a significant fact about the data.

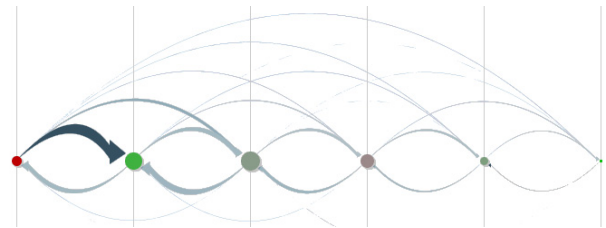


Figure 9. *Anomalous transitions: In this data set, each of the arcs spanning more than two columns represents a surprising data point.*

### Pilot 2

Liz, a social network analyst and researcher, had recently analyzed data on communication patterns in a community of 146 people within a large company. Each person in the community was classified on five dimensions, such as corporate division and geographic location. The network had directed edges, with an edge from person A to person B meaning that person A had reported communicating recently with person B. Liz used both UCINET/NetDraw [6] and Microsoft Excel to study the data. An anonymized screenshot of one view of her data in PivotGraph is shown in Figure 10.

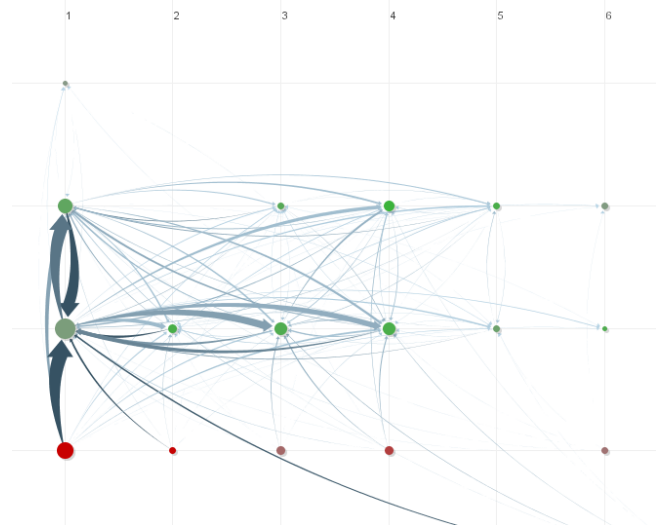


Figure 10. *Communication network of people in a large company. X-axis is division, y-axis is office geography. The division in the leftmost column has far more cross-location communication than the others.*

As with other users, when using PivotGraph she pointed out one and two-variable comparisons as she used the software. Inspecting a roll-up on the company division variable, she said, "There is more communication between Division X and Division Y than between Division X and Division Z."

She also pointed to examples of communication asymmetry which she said she had not noticed in previous analyses. (The underlying graph, based on recent communication between two people, could have been expected to be symmetric since communication is a two-way relationship; thus asymmetry indicated an interesting lack of reciprocity in people's reporting of the communication.) She compared the tool favorably to sorted matrix views which she had also tried, but which required “hunting” to make these comparisons. She also pointed to the varying widths of the edges—in particular, the very small widths of low-weight edges—as helpful for keeping the graph relatively uncluttered.

Liz said she like the fact that the program could automatically aggregate nodes, since she had sometimes done this by hand in NetDraw. One unexpected aspect of Liz's use of the tool was to roll up her graph on two variables, A and B, and then use the selection menus to cycle through values of A, allowing her to rapidly switch focus between different one-dimensional slices. Liz also suggested some additional features for the program. She pointed out that it would be helpful to have an option to manually remove certain distracting nodes. (NetDraw, like many other graph diagram tools, offers this feature.) She also noted that the automatic axis ordering occasionally interfered with the natural ordering of an axis, and asked for a manual override of this feature.

### **Pilot 3**

Michael, an organizational change consultant, offered to share anonymized data from an engagement with a company that had recently undergone a merger. The data described communication patterns among employees of the company, with each employee classified according to five different dimensions (e.g., gender, department, and which of the two merging companies the person had come from). Michael had previously used InFlow [17] to study the data.

Using PivotGraph, Michael made comparisons involving different patterns of communication between men and women, which he said he found particularly interesting. He said his favorite way to use the tool was to keep one axis tied to a given variable and skip between other variables on the other axis.

Michael had not previously used a tool that automatically aggregated nodes on multiple dimensions. He said that this made the tool unsuitable for some analyses, but that it would provide new insights on intergroup dynamics. He indicated he would like to use the program again; it turned out that he was sufficiently motivated to do so that he wrote a custom data converter to transform his InFlow data into a format readable by PivotGraph.

### **Summary of Pilots**

The pilots provide preliminary evidence that PivotGraph may be a helpful alternative view of multivariate graph data. Each of the five users had spent significant time

looking at their data with other tools, but claimed to spot new patterns using PivotGraph and all expressed a desire to use the software again. All pointed to the ability to make comparisons across variables as an important feature, and all exploited the interactivity to look at many different views of the data.

The pilot tests made certain limitations clear as well. In particular, several users noted that PivotGraph would be best as a complement, rather than a replacement, for current tools due to its emphasis on aggregate values and lack of support for viewing graph topology.

### **FUTURE DIRECTIONS**

The PivotGraph technique lends itself to a number of interesting potential extensions. Several users have suggested that it would be helpful to compare more than two dimensions at once. An obvious method to try would be a 3D representation, in which the *z*-axis could correspond to a third categorical property. Some initial experimentation indicates that labeling and occlusion will present difficult challenges, but they may well be solvable. It may also be possible to present extra dimensions of data by drawing several graph diagrams at once, following Tufte's “small multiples” concept [23].

The PivotGraph approach has a certain inherent scalability because it relies on summarizing graphs. A graph with a 1,000,000 nodes can be visually tractable when rolled up on just two categories with a small number of values each. Performance in summarization may still be an issue, however. When there are many possible categories, it might be helpful if the computer could recommend particular combinations—a kind of graph-based projection pursuit. In addition, with more complex data sets, sophisticated dynamic query interfaces may be needed.

Another natural direction is to visualize other types of attributes. This paper has focused on discrete-valued dimensions, but it would be interesting to investigate extensions to dimensions with continuous values; the challenge here is that projection onto real-valued dimensions is much less likely to create a simplified graph with a reduced number of nodes and edges. It would also be helpful to display graphs in which edges are multivariate as well as nodes. In our experience graphs with multivariate edges are somewhat less common in practice than ones with multivariate nodes, but they do arise. In social network analysis, for instance, it is often helpful to characterize the relationships between people along several dimensions. Extending the pivot mechanism to edge types as well as node types poses interesting visualization and interaction challenges.

### **CONCLUSION**

The PivotGraph system is a visualization tool for exploring and analyzing multivariate graphs. By combining a grid-based visualization with straightforward data reduction techniques modeled on OLAP, it allows users to interact



with a series of simple, clarified views of a complex graph. These views are designed for generating hypotheses about how connections relate to multiple attributes.

Early tests with expert analysts suggest that the visualizations in PivotGraph may provide a different perspective from traditional graph visualization tools, which could potentially help people find previously unnoticed patterns. Finally, this paper has outlined a number of promising directions for future research that build on the simple data and interaction model used by PivotGraph.

#### ACKNOWLEDGMENTS

The PivotGraph system has benefited from the design suggestions and help of many people: Li-Te Cheng, Kate Ehrlich, Dan Forno, Deepika Kakrania, Michael Martin, John Patterson, and Mark Squillante. This paper has been improved greatly by comments from Steve Rohall, Fernanda Viegas, and the anonymous referees.

#### REFERENCES

1. Auber, D. Tulip: A Huge Graph Visualisation Framework. In P. Mutzel and M. Junger, eds., *Graph Drawing Software, Mathematics and Visualization*, Springer-Verlag, 2003.
2. Batagelj, V., and Mrvar, A. Pajek - Analysis and Visualization of Large Networks. In P. Mutzel and M. Junger, eds., *Graph Drawing Software, Mathematics and Visualization*, Springer-Verlag, 2003.
3. Battista, G. d., P. Eades, et al. (1999). *Graph Drawing: Algorithms for the Visualisation of Graphs*, Prentice Hall.
4. Becker, R., S. Eick, A. Wilks. Visualizing Network Data. *IEEE Transactions on Visualization and Computer Graphics*, 1 (1995).
5. Bollobás, B. *Modern Graph Theory*. Springer-Verlag, 1998.
6. Borgatti, S.P., Everett, M.G. and Freeman, L.C. *Ucinet for Windows: Software for Social Network Analysis*. Harvard, MA: Analytic Technologies. 2002.
7. Codd, E. F., S. B. Codd, C. T. Salley. Beyond Decision Support. *Computerworld*, 27 (30): 87-89. 1993.
8. Fekete J.-D., The InfoVis Toolkit, *IEEE Symposium on Information Visualization* (2004),
9. Fekete, J.-D., D. Wang, et al. (2003). Overlaying Graph Links on Treemaps. *IEEE Information Visualization 2003 Symposium Poster*
10. Freeman, L. Visualizing Social Networks. *Journal of Social Structure*, 1, 2000 (1).
11. Ghoniem, M., J.-D. Fekete, P. Castagliola. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. *IEEE Symposium on Information Visualization* (2004).
12. Gray, J., Bosworth, A., Layman, A., and Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. *ICDE 1996*: 152-159.
13. van Ham, F. and van Wijk, J. Interactive Visualization of Small World Graphs. *IEEE Symposium on Information Visualization* (2004).
14. Herman, I., G. Melancon, M. Scott Marshall. Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 6 (2003), 24-43.
15. Kaufmann, M. and D. Wagner, Eds. (2001). *Drawing Graphs, Methods and Models*. Lecture Notes in Computer Science, Springer.
16. Kerr, Bernard. ThreadArcs. *IEEE Symposium on Information Visualization* (2003)
17. Krebs, V. E. *InFlow, version 3.0*, 2002. Cleveland: Orgnet.com
18. *NetMiner, version 2.5*. Cyram Co. Ltd., 2004. <http://www.netminer.com/>
19. *Spotfire DecisionSite*, Spotfire, Inc., Somerville, Massachusetts, 2005. <http://www.spotfire.com>
20. Stolte, C., Tang, D. and Hanrahan, P. Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 8, No. 1, January 2002.
21. Sugiyama, K., S. Tagawa, M. Toda. Methods for Understanding Hierarchical System Structures. *IEEE Transactions on Systems, Man and Cybernetics* 11 (2), 1981.
22. *Tableau 1.5*, Tableau Software, Seattle, Washington, 2005. <http://www.tableausoftware.com>
23. Tufte, Edward. *Envisioning Information*. Graphics Press, 1990.
24. Wattenberg, M. Arc Diagrams: Visualizing Structure in Strings. *IEEE Symposium on Information Visualization* (2002).